

UNCLASSIFIED

AD NUMBER

ADB003065

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; MAR 1975. Other requests shall be referred to Air Force Weapons Laboratory, Attn: DYS, Kirtland AFB, NM 87117.

AUTHORITY

AFWL ltr, 29 Apr 1985

THIS PAGE IS UNCLASSIFIED

AD B003065

AUTHORITY: AFWL 17, 29 Apr 85



L
AD B 003065

SIMULATION BRANCH PLOTTING AND UTILITY ROUTINES I

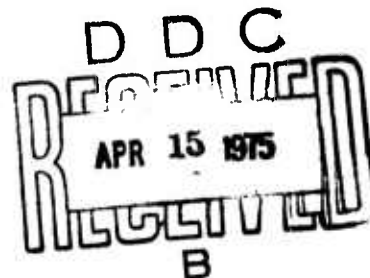
James D. Letterio, Sgt, USAF
Clifford E. Rhoades, Jr., Lt, USAF
Robert W. Boyd, Capt, USAF

March 1975

Final Report for Period October 1973 - June 1974

Distribution limited to US Government agencies only because of
test and evaluation of preliminary information (Mar 1975). Other
requests for this document must be referred to AFWL (DYS), Kirtland
AFB, NM 87117.

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117



This final report was prepared by the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico, under Job Order 88091701. Lt Clifford E. Rhoades, Jr. was the Laboratory Project Officer-in-Charge.

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

Clifford E. Rhoades, Jr.

CLIFFORD E. RHOADES, JR.
Lt, USAF
Project Officer

Thomas C. May

THOMAS C. MAY
Major, USAF
Chief, X-Ray Simulation Branch

FOR THE COMMANDER

David M. Ericson, Jr.

DAVID M. ERICSON, JR.
Lt Colonel, USAF
Chief, Technology Division

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWL-TR-74-198	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SIMULATION BRANCH PLOTTING AND UTILITY ROUTINES I		5. TYPE OF REPORT & PERIOD COVERED Final Report: October 1973 - June 1974
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) James D. Letterio, Sgt, USAF Clifford E. Rhoades, Jr., Lt, USAF Robert W. Boyd, Capt, USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Weapons Laboratory (DYS) Kirtland AFB, NM 87117		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62601F 8809 1701
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Weapons Laboratory (DYS) Kirtland AFB, NM 87117		12. REPORT DATE March 1975
		13. NUMBER OF PAGES 44
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Distribution limited to US Government agencies only because of test and evaluation of preliminary information (Mar 1975). Other requests for this document must be referred to AFWL (DYS), Kirtland AFB, NM 87117.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Plotting package 2-D magnetohydrodynamics Microfilm BRAHMA FTN (Fortran IV Extended) SCOPE 3.2		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A set of plotting and utility routines written for use with the 2-dimensional magnetohydrodynamics code BRAHMA is described. All routines are FTN com- patible and were written for use under the SCOPE 3.2 operating system.		

UNCLASSIFIED

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SUMMARY

This report briefly describes a set of plotting and utility routines written for use with the two-dimensional magnetohydrodynamics code BRAHMA.

Section I is concerned with microfilm plotting routines. Section II is primarily concerned with utility routines, the exception being SCRIBE, a calcomp and microfilm character plotting routine.

All routines are FTN compatible and were written for use under the SCOPE 3.2 operating system.

PREFACE

The authors would like to thank SSgt Barry Miller for providing RQUEST. They are indebted to SSgt Richard Vore for the original DYDIM and Loren Meissner for the Hershey character set in SCRIBE.

CONTENTS

<u>Section</u>		<u>Page</u>
I	MICROFILM PLOTTING ROUTINES	5
II	UTILITY ROUTINES AND SCRIBE	27
	DISTRIBUTION	43

ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	Example of Line and Point Plotting	10
2	Example of MAPG Grid	12
3	Example of MAPGSL Grid	13
4	Example of Direction Field via PLOTV	16
5	Example of Computer Output Processed by SECURE	40
6	Example of Summary Page as Given by SECURE	41
7	Example of Dayfile Message as Given by SECURE	42

SECTION I
MICROFILM PLOTTING ROUTINES

ADJØST--Convert Arguments

Summary: Converts nonlinear arguments to linear before scaling. This routine is not usually called directly.

Vital Statistics: Entry points: ADJØST

Routines called: ALOG1Ø

Common blocks: SMAP

ASSIGN--Connect Logical Number and I/O Device

Summary: Subroutine ASSIGN is used to establish a relationship between a logical unit number and a specific input or output tape volume.

Calling sequence: CALL ASSIGN (L,H)

where: L is the (integer) logical unit number, in the range 1-99.

H is the tape volume identification, that is the volume serial number in R5 format.

Input arguments: L,H

Output arguments: (none)

Vital Statistics: Entry points: ASSIGN

Routines called: REMARK, RECALL, CPAREA, ØØUNIP, RQUEST

CLOCK--Obtain Alphanumeric Time and Date

Summary: CLOCK allows the user to obtain the alphanumeric values of the current time and date.

Calling sequence: CALL CLOCK (A,B)

where: A Current reading of system clock as returned by SCOPE.

B date as returned by SCOPE.

Input arguments: (none)

Output arguments: A,B

Vital Statistics: Entry points: CLOCK

Routines Called: TIME, DATE

Common blocks: (none)

CNTRL--Obtain Job Information

Summary: CNTRL obtains the following information from the control point area and places it in the common block INFO: Jobname, CP time limit, CP time, PP time, Priority, CM field length, ECS field length, the initial CM field length and the initial ECS field length. This routine is not usually called directly.

Calling sequence: CALL CNTRL
 Input arguments: (none)
 Output arguments: (none-results placed in common block INFO)
 Vital statistics: Entry points: CNTRL
 Routines called: CPAREA
 Common blocks: INFO, INFOB

CPAREA--Obtain Control Point Area

Summary: CPAREA obtains the control point area from the system. This routine is not usually called directly.

Calling sequence: CALL CPAREA(CPINFO)
 where: CPINFO is an array of 128 locations into which is written the control point area.
 Input arguments: (none)
 Output arguments: CPINFO
 PP routine
 Remarks: 1. CXA is used to obtain the control point area.
 2. CPAREA is written in COMPASS.
 Vital statistics: Entry points: CPAREA

CRTID--Initialize 80

Summary: CRTID should be called once and only once in any job producing CRT output and must precede all other calls to CRT routines. It produces at the beginning of the output two blank frames and a title page on which is written the CRT job identification, the date and time when the output was generated, and the classification (if any) of data being output. The classification also appears at the top and bottom of every page of output.

Calling sequence: CALL CRTID (A,I,N)
 CALL CRTID (A,I)
 CALL CRTID (A)
 where: A unused
 I is the classification.
 I=0 for classified information
 I=1 for unclassified information.
 N unused
 Input arguments: A,I,N
 Output arguments: (none)
 Remarks: 1. Only one copy of the output from a single job can be produced.
 2. Common block CLSCOM (one word) must be defined in the (0,0) overlay.
 Vital statistics: Entry points: CRTID
 Routines called: PLOTQ, SRDID, NUMARG
 Common blocks: CLSCOM

DEVICE--Dummy Routine

Summary: Subroutine DEVICE is a dummy routine and exists only for compatibility with previous versions of the plot package.

EMPTY--Empty Output Buffer

Summary: EMPTY empties the output buffer.

Calling sequence: CALL EMPTY (N)
 where: N Fortran unit number
 Input arguments: N
 Output arguments: (none)
 Vital statistics: Entry points: EMPTY
 Routines called: ENDFILE, BACKSPACE

FRAME--Advance the Film

Summary: FRAME will advance the film N frames. If N is omitted, the film is advanced one frame. A classification label is written on the frame if necessary.

Calling sequence: CALL FRAME (N)
 CALL FRAME
 where: N is the number of frame advances desired.
 Input arguments: N (or none)
 Output arguments: (none)
 Remarks: (none)
 Vital statistics: Entry points: FRAME
 Routines called: PLOTQ, SRDID
 Common blocks: CLSCOM

GRIDG--Plot a Grid

Summary: Plots a full grid in the CRT plane. This routine is usually not called directly. (see MAPG.)

Vital statistics: Entry points: GRIDG
 Routines called: ALOG10, PLOTQ
 Common blocks: KEEP

INTRPL8--Linear Interpolation

Summary: This routine is not usually called directly.

Vital statistics: Entry points: INTRPL8
 Routines called: SQRT
 Common blocks: SMAP

LINE--Draw a Line

Summary: LINE will sweep a line from (X1, Y1) to (X2, Y2) with intensity I.

Calling sequence:

```
CALL LINE (X1, Y1, X2, Y2)
CALL LINE (X1, Y1, X2, Y2, I)
```

where:

X1, Y1 } are the coordinates of the initial point.
 X2, Y2 } are the coordinates of the terminal point.
 I specifies the intensity.
 I=0 for low intensity
 I=1 for high intensity.

Input arguments: X1, Y1, X2, Y2, I

Output arguments: (none)

Remarks:

1. LINE may be called with 4 or 5 arguments.
2. If I is omitted, the current intensity will be used.
3. See fig. 1 for example.

Vital statistics: Entry points: LINE
 Routines called: ALOG10, NUMARG, PLOTQ
 Common blocks: SMAP, KEEP

MAP--Establish Linear or Logarithmic Mapping

Summary: MAP (and its entry points MAPLL, MAPSL, and MAPLS) will establish a mapping from the rectangle in the X-Y plane with corners (XMIN, YMIN), (XMAX, YMAX) onto the rectangle in the CRT plane with corners (XMI, YMI), (XMA, YMA). Each axis may have a linear or logarithmic scale.

Calling sequence:

```
CALL MAP (XMIN, XMAX, YMIN, YMAX)
CALL MAP (XMIN, XMAX, YMIN, YMAX, XMI, XMA, YMI, YMA)
CALL MAPLL similarly
CALL MAPSL similarly
CALL MAPLS similarly
```

where:

XMIN, XMAX } are the user's minimum and maximum
 YMIN, YMAX } Cartesian coordinates.
 XMI, XMA } are the minimum and maximum coordinates,
 YMI, YMA } between 0. and 1., of the desired portion
 of the CRT plane.

Input arguments: XMIN, XMAX, YMIN, YMAX XMI, XMA, YMI, YMA

Output arguments: (none)

Remarks:

1. The linear-logarithmic scaling is determined for the X and Y axes, as follows:

	X axis	Y axis
MAP	linear	linear
MAPLL	log	log
MAPSL	linear	log
MAPLS	log	linear

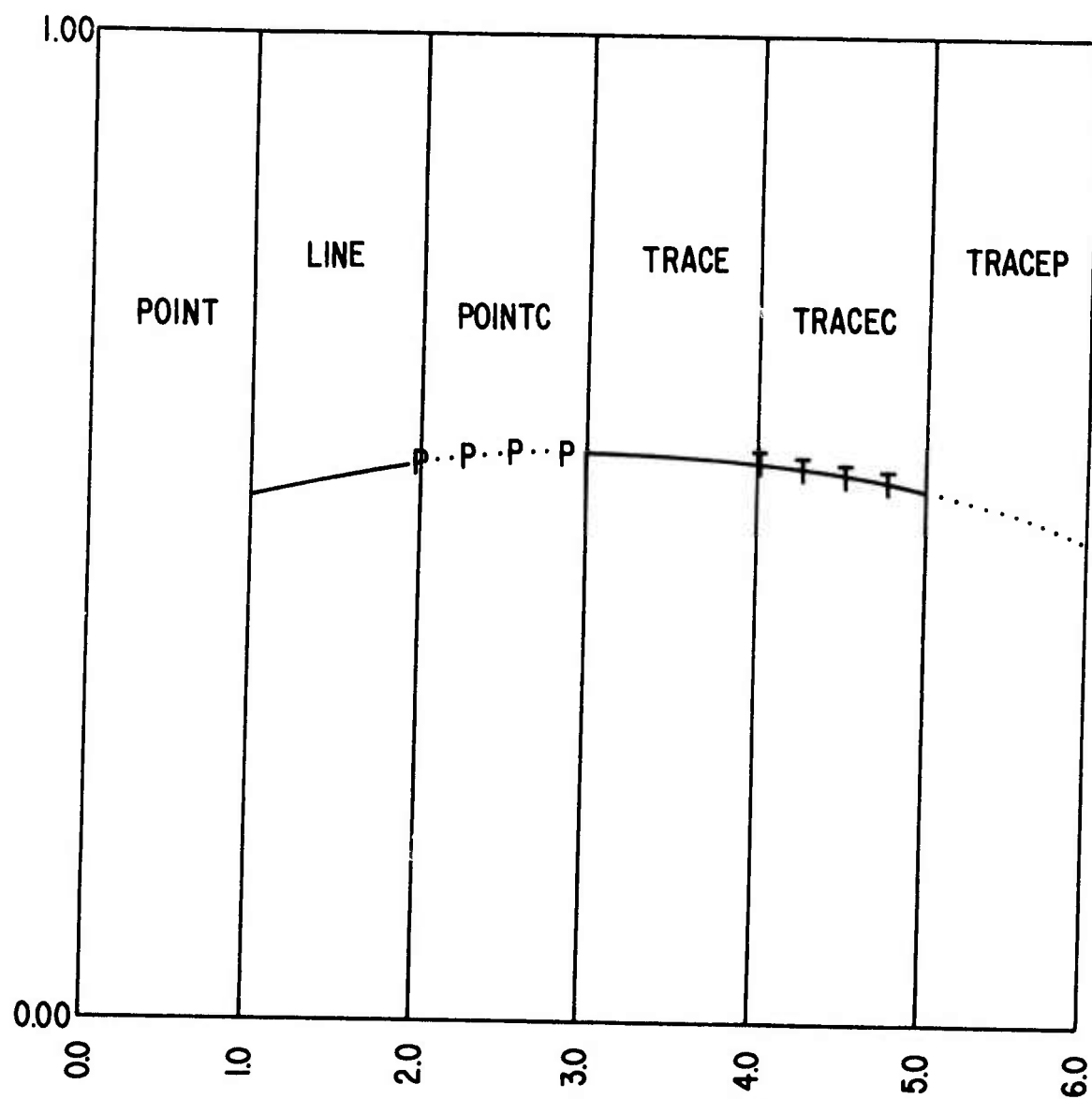


Figure 1. Example of Line and Point Plotting

2. Unless reset, this mapping applies to all subsequent plotting, except character plotting controlled by SETCH.
3. The routines may be called with four or eight arguments. If the last four arguments are omitted, $XMI=YMI=0.$ and $XMA=YMA=1.$ are used.
4. $0. \leq XMI \leq XMA \leq 1.$ and $0. \leq YMI \leq YMA \leq 1.$
5. $XMIN > 0.$ and $YMIN > 0.$ for logarithmic scaling.

Vital statistics: Entry points: MAP, MAPLL, MAPSL, MAPLS
 Routine called: ALOG10, NUMARG
 Common blocks: SMAP

MAPG--Establish Mapping with Full Grid

Summary: MAPG (and its associated routines MAPGLL, MAPGSL, AND MAPGLS) will establish a mapping from the rectangle in the X-Y plane with corners (XMIN, YMIN), (XMAX, YMAX) onto the rectangle in the CRT plane with corners (XMI, YMI), (XMA, YMA). Each axis may have a linear or logarithmic scale. A full reference grid with scale numbers will be drawn (see figs. 2 and 3).

Calling sequence:

CALL MAPG (XMIN, XMAX, YMIN, YMAX)
 CALL MAPG (XMIN, XMAX, YMIN, YMAX, XMI, XMA, YMI, YMA)
 CALL MAPGLL similarly
 CALL MAPGSL similarly
 CALL MAPGLS similarly

where:

XMIN, XMAX } are the user's minimum and maximum
 YMIN, YMAX } Cartesian coordinates.
 XMI, XMA } are the minimum and maximum coordinates,
 YMI, YMA } between 0. and 1., of the desired portion
 of the CRT plane.

Input arguments: XMIN, XMAX, YMIN, YMAX, XMI, XMA, YMI, YMA

Output arguments: (none)

Remarks: 1. The linear-logarithmic scaling is determined for the X and Y axes as follows:

	X axis	Y axis
MAPG	linear	linear
MAPGLL	log	log
MAPGSL	linear	log
MAPGLS	log	linear

2. Unless reset, this mapping applies to all subsequent plotting, except character plotting controlled by SETCH.
3. The routines may be called with 4 or 8 arguments. If the last four arguments are omitted, $XMI=YMI=0.11328$ and $XMA=YMA=1.0$ are used.
4. $0. \leq XMI \leq XMA \leq 1.$ and $0. \leq YMI \leq YMA \leq 1.$
5. $XMIN > 0.$ and $YMIN > 0.$ for logarithmic scaling.

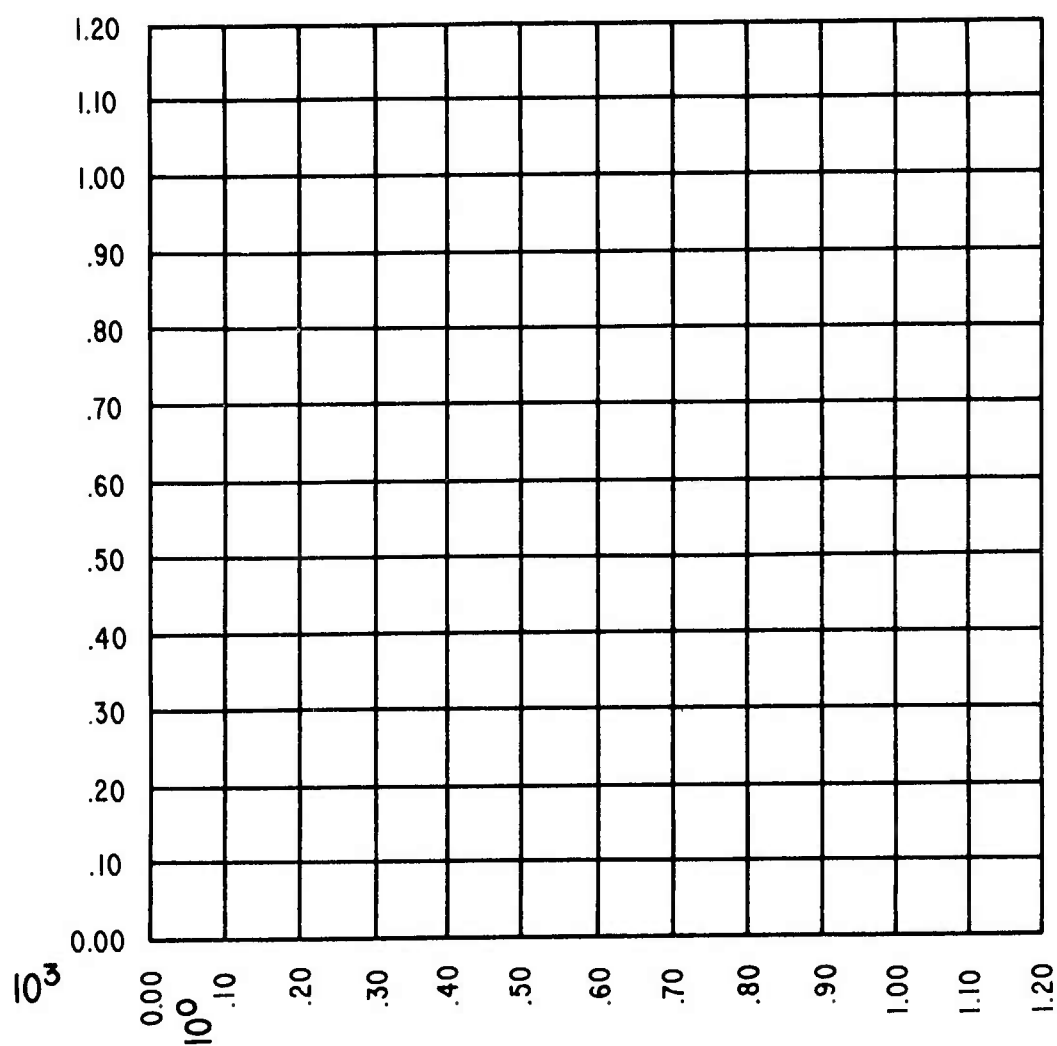


Figure 2. Example of MAPG Grid

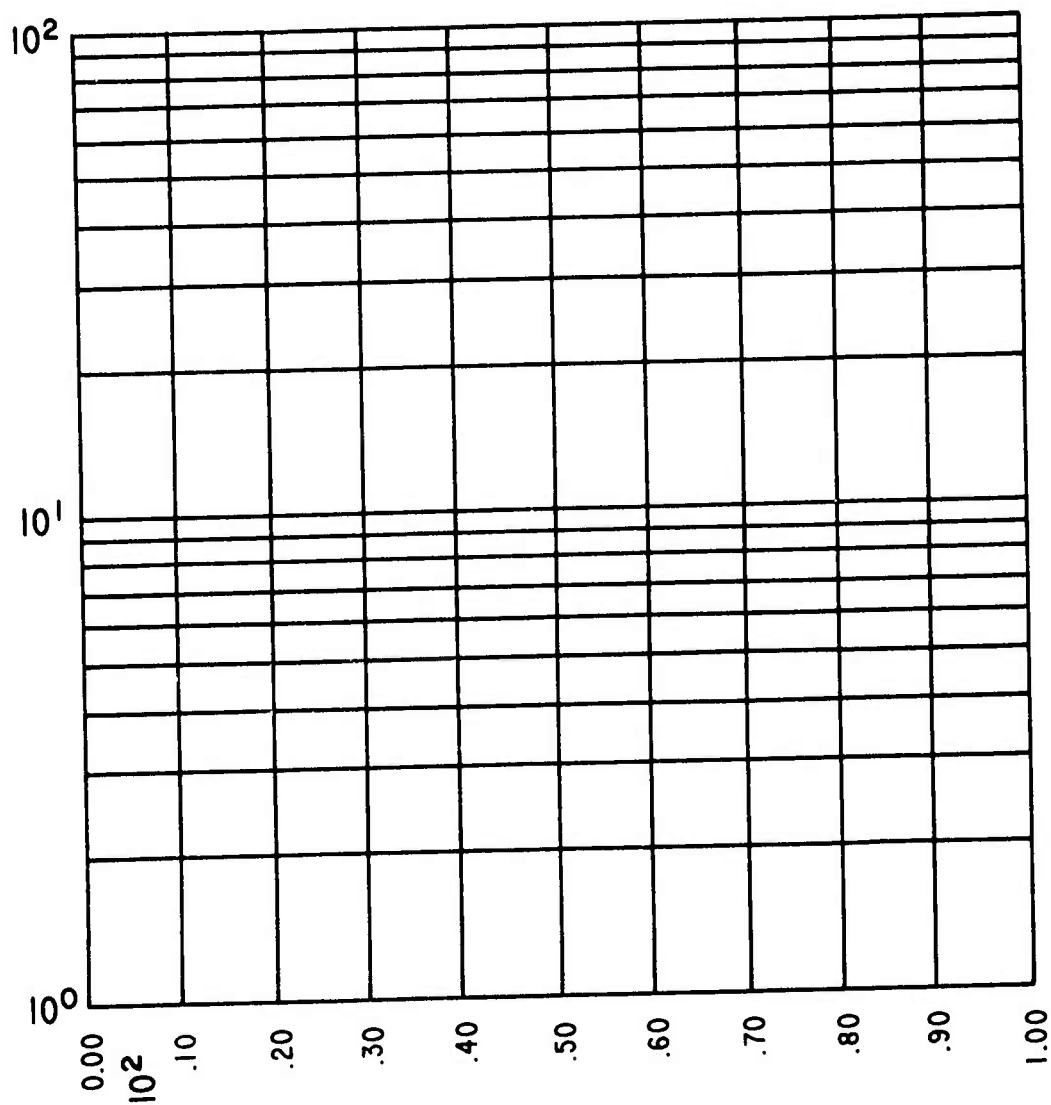


Figure 3. Example of MAPGSL Grid

Vital statistics: Entry points: MAPG, MAPGSL, MAPGLS, MAPGLL
 Routines called: ALOG10, GRIDG, NUMARG
 Common blocks: SMAP

NUMARG--Number of Arguments

Summary: Counts the number of arguments in the calling sequence. This routine is not usually called directly.

Vital statistics: Entry points: NUMARG
 Routines called: (none)
 Common blocks: (none)

NOTE: NUMARG is coded in COMPASS and is FTN compatible only.

O0TIM--Obtain Time Interval, Time Limit, Time Remaining, and Priority

Summary: O0TIM allows the user to time various phases of his program and to obtain the current value of his time limit and the time remaining.

Calling sequence: CALL O0TIM (I, J, K, L)
 CALL O0TIM (I, J)
 CALL O0TIM (I)
 CALL O0TIM

where: I is the central processing time in integer microseconds charged since the last call to O0TIM.
 J is the current time limit, in integer microseconds.
 K is the current value of the time remaining, in integer microseconds.
 L is current priority in integer.

Input arguments: (none)

Output arguments: I,J,K,L

Remarks: 1. O0TIM can be called with 0,1,2,3, or 4 arguments. The call without arguments is used to initialize the time.
 2. With overlays common block O0TIMC must be defined with one location in the (0,0) overlay.

Vital statistics: Entry points: O0TIM
 Routines called: NUMARG, CNTRL
 Common blocks: O0TIMC, INFO

O0UNIT--Tape Volume and Physical Unit Correspondence

Summary: Subroutine O0UNIT provides a method whereby the physical unit number of the tape drive to which a given volume is (or was) assigned may be determined.

Calling sequence: CALL O0UNIT (IVOL, ILUN)

where: IVOL is the volume serial number in R5 format.
 ILUN is the returned physical unit number in R2 format.

Input argument: IVOL
 Output argument: ILUN
 Vital statistics: Entry points: OØUNIT, OØUNIP
 Common blocks: OØUCOM (20 words)

Remarks: 1. All tapes must be assigned via calls to ASSIGN.
 2. If a return has been done on a given tape, that tape may not be currently mounted even though ILUN is returned with a value other than 2RNO.
 3. Common block OØUCOM must be defined in the (0,0) overlay and all 20 elements DATAed to zero.

PLOTV--Draw an Arrow

Summary: PLOTV will draw an arrow from (X1,Y1) to (X2,Y2).

Calling sequence:

CALL PLOTV (X1,Y1,X2,Y2)
 CALL PLOTV (X1,Y1,X2,Y2,H)

where: X1, Y1 } are the coordinates of the initial point.
 X2, Y2 } are the coordinates of the final point.
 H is the size of the arrowhead, between 0. and 1.

Input arguments: X1,Y1,X2,Y2,H
 Output arguments: (none)
 Remarks: 1. The routine may be called with 4 or 5 arguments. (see fig. 4 for example.)
 2. $0 \leq H \leq 1$. If H is omitted, the length of the arrowhead will be proportional to the length of the vector.

Vital statistics: Entry points: PLOTV
 Routines called: ADJØST, NUMARG, PLOTQ, SQRT
 Common blocks: SMAP, KEEP

POINT--Plot a Single Point

Summary: POINT will plot a point at (X,Y) with intensity I.

Calling sequence:

CALL POINT (X,Y)
 CALL POINT (X,Y,I)

where: X,Y are the coordinates of the point.
 I specifies the intensity.
 I=0 for low intensity.
 I=1 for high intensity.

Input arguments: X,Y,I
 Output arguments: (none)
 Remarks: 1. POINT may be called with 2 or 3 arguments.

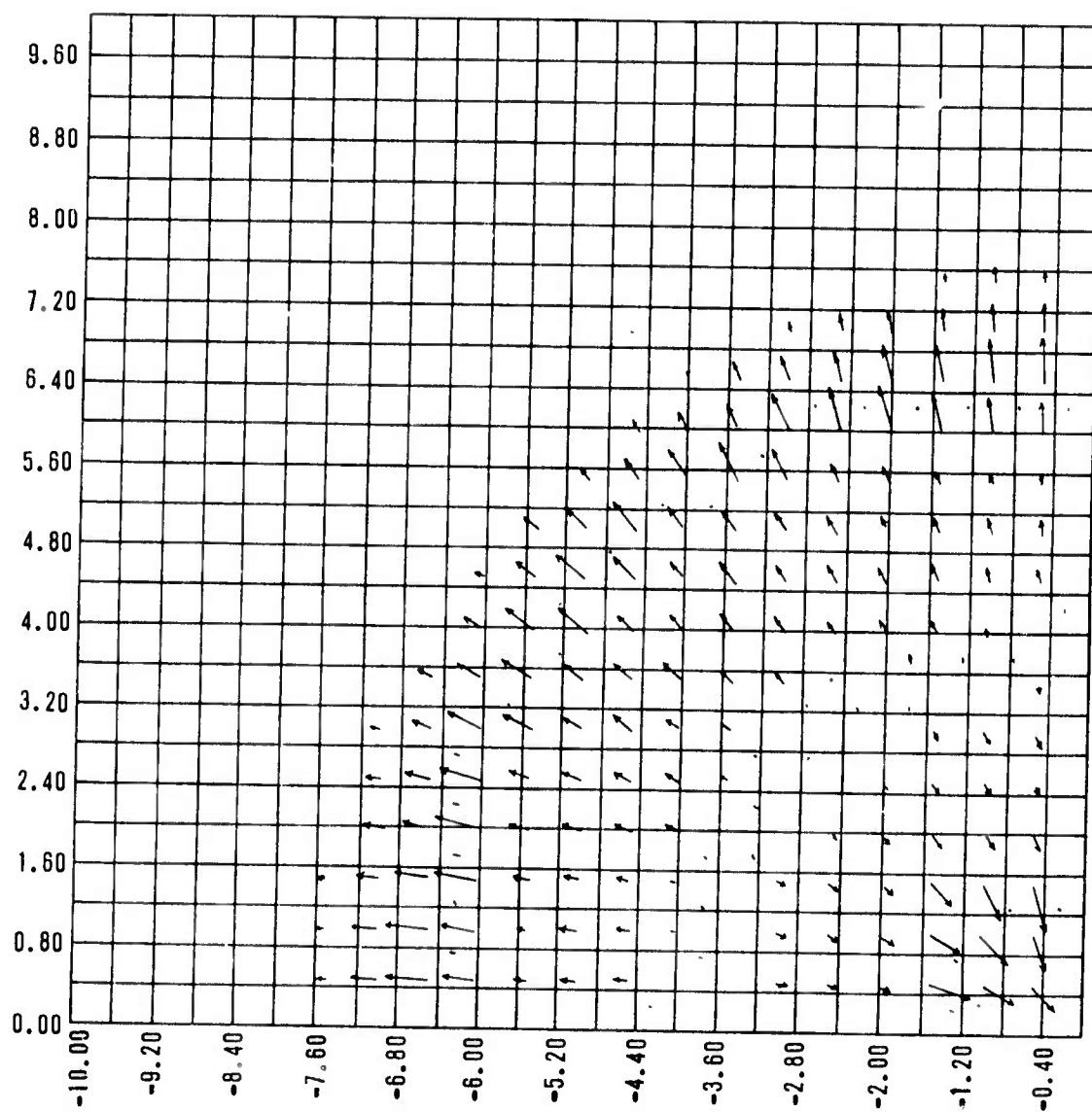


Figure 4. Example of Direction Field via PLOTV

2. If I is omitted, the current intensity setting is used.
 3. See fig. 1 for example.

Vital statistics: Entry points: POINT
 Routines called: ALOG10, NUMARG, PLOTQ
 Common blocks: SMAP, KEEP

POINTC--Plot Points with Label

Summary: POINTC will plot the N points given by X and Y with the specified 280 character superimposed on points more than K raster points apart.

Calling sequence:

CALL POINTC (1HA,X,Y,N)
 CALL POINTC (1HA,X,Y,N,IX)
 CALL POINTC (1HA,X,Y,N,IX,IY)

where: A is a single Hollerith character.
 X,Y are the names of floating point arrays of X and Y coordinates.
 N is the number of points to be plotted.
 IX, IY are the increments between successive storage locations in X and Y, respectively.

Input arguments: A,X,Y,N,IX,IY
 Output arguments: (none)
 Remarks: 1. POINTC may be called with 4, 5, or 6 arguments.
 2. The data point will use the current intensity.
 3. The value of K and the size, case, intensity, and type face of the character may be specified in SETPCH.
 4. IX and IY will be assumed 1 if they are omitted.
 5. The first argument may be the name of a variable.
 6. See fig. 1 for example.

Vital statistics: Entry points: POINTC
 Routines called: ALOG10, NUMARG, PLOTQ, SQRT
 Common blocks: SMAP, KEEP

Q7PLOT

Summary: This routine is not usually called directly.

Vital statistics: Entry points: Q7PLOT
 Routines called: Q8PLOT
 Common blocks: SMAP, Q7QUAD

Q8PLOT

Summary: This routine is not usually called directly.

Vital statistics Entry points: Q8PLOT
 Routines called: TRACE, TRACEP
 Common blocks: Q7QUAD

RCONTR--Contour Plot, Rectangular Grid

Summary: RCONTR requires a rectangular coordination of the X,Y-plane, as specified by the one-dimensional arrays X and Y. If FCN is the function whose level curves (contours) are to be drawn, RCONTR assumes that

$$A(I,J) = FCN(X(I), Y(J)).$$

Calling sequence: CALL RCONTR (K1, C, K2, A, MAX, X, IMIN, IMAX, ISTEP, Y, JMIN, JMAX, JSTEP)

where:

- K1,K2 integer variables determining which plotting option is to be used. (See options below.)
- C one-dimensional floating point array of test values for level curves. (See options below.)
- A Two-dimensional floating point array of function values.
- MAX integer variable, maximum row dimension (first number in the DIMENSION statement for A).
- X,Y one-dimensional floating point arrays defining a rectangular grid in the X,Y-plane.
- IMIN Integer variable, minimum I-value.
- IMAX Integer variable, maximum I-value.
- ISTEP Integer variable, increment between successive I-values.
- JMIN, } corresponding information for J-values.
- JMAX, }
- JSTEP }

Input arguments: K1,C,K2,A,MAX,X,IMIN,IMAX,ISTEP,Y,JMIN,JMAX,JSTEP

Output arguments: C (if K1 < 0)

Vital statistics: Entry points: RCONTR
 Routines called: Q7PLOT, SETCRT
 Common blocks: Q7QUAD

Options: The first three arguments (K1, C, K2), of RCONTR determine the level lines (contour values) to be plotted. Three plotting options are available, depending on the value of K1.

+OPTION If $K1 > 0$, then the array C contains K1 numbers which are to be used as test values for plotting the level lines
 $FCN(X,Y)=C(I)$.
 Those contours $C(I)$ with $I < K2$ plotted as a series of dots, while those with $I \geq K2$ are plotted as solid lines.

-OPTION If $K1 < 0$, then $|K1|$ equally-spaced test values between $C(1)$ and $C(2)$ (inclusive) are generated and placed in C as output. The numbers are stored in order, with $C(1)$ remaining unchanged and new $C(|K1|) = \text{old } C(2)$. Plotting then proceeds as in the + option.

0 OPTION If $K1=0$, then $FO=C(1)$, $DF=C(2)$, and all level curves of the form

$$FCN(X,Y)=FO + M*DF, M \text{ an integer,}$$

which pass through the grid are plotted. Those contours less than $C(1)$ are plotted as a series of dots, while those greater than or equal to $C(1)$ are plotted as solid lines. In this case, $K2$ is ignored.

(Note: an error exit is taken if $C(2) \leq 0$ in the 0 option.)

RECALL--Enter Recall

Summary: RECALL places the code in recall until the next monitor loop.
This routine is not usually called directly.

Calling sequence: CALL RECALL
Input arguments: (none)
Output arguments: (none)
Remarks: 1. RECALL is written in COMPASS
Vital statistics: Entry points: RECALL

RQUEST--Run Time Tape Request

Summary: RQUEST requests a preassigned tape to be assigned to the job at run time. The tape must either be preassigned or mounted and labeled. This routine is not usually called directly.

Calling sequence: CALL RQUEST (IUNIT, 2HDN, 2HIP, 2HLB)
where: IUNIT is the FORTRAN I/O unit number
DN specifies the tape density and is:
MT (default).
HI 556.
HY 800.
LO 200.
IP specifies the recording format and is.
X External format (not available under SCOPE 3.4).
S Stranger format.
L Long format. (not available under AFWL SCOPE 3.4)
SC Scope Standard.
LB specifies the labeling and is:
E Prelabeled ANSI.
N ANSI label to be written.
Y 3000 series label.
NO Unlabeled.

Input arguments: (all four)
Output arguments: (none)
Remarks: 1. This routine was written by SSgt Barry Miller in COMPASS.
2. This routine must be modified to supply VSN and file header information for SCOPE 3.4.
Vital statistics: Entry points: RQUEST

SETCH--Set Plot Mode

Summary: SETCH specifies the starting character position (X,Y), intensity, case, size, orientation, and type face for the next line plotted by WOT100. X and Y refer to an absolute position on the CRT face.

Calling sequence: CALL SETCH
 CALL SETCH (X,Y)
 CALL SETCH (X,Y,I)
 CALL SETCH (X,Y,I,IC)
 CALL SETCH (X,Y,I,IC,IS)
 CALL SETCH (X,Y,I,IC,IS,IO)
 CALL SETCH (X,Y,I,IC,IS,IO,IT)

where: X,Y are the coordinates of the center of the first character (see Remarks 3 and 4).
 I specifies the intensity.
 I=0 for low intensity.
 I=1 for high intensity.
 IC specifies the case.
 IC=0 for upper case(normal).
 IC=1 for lower case.
 IS specifies the size.
 IS=0 for miniature characters.
 IS=1 for small characters.
 IS=2 for medium characters.
 IS=3 for large characters.
 IO specifies the orientation.
 IO=0 for horizontal orientation, the line running left to right.
 IO=1 for vertical orientation, the line running bottom to top.
 IT specifies the type face.
 IT=0 for Roman type.
 IT=1 for italic type.

Input arguments: X,Y,I,IC,IS,IO,IT

Output arguments: (none)

Remarks:

1. SETCH may be called with 0,2,3,4,5,6, or 7 arguments.
 CAUTION: The call without arguments is used to restore current SETCH parameters from before the last SETLCH call and should not normally be made.
2. I,IC,IS,IO, and IT will be zero initially. Arguments which are omitted will not be reset.
3. In this routine, the CRT plane is considered to be a grid of rectangles, each containing one character of the chosen size. The number and dimensions of these rectangles depend on the character size and orientation.

The X and Y coordinates here refer to these rectangles, and hence they also depend on character size and orientation. X and Y may have fractional parts, so that a character need not be plotted in the center of one of these rectangles. The following table gives the dimensions of this grid for each size character.

<u>SIZE</u>	<u>CHARACTERS</u>	<u>LINES</u>
0(miniature)	128	64
1(small)	85	42
2(medium)	64	32
3(large)	42	21

4. For an orientation of 0, X refers to the character position, and Y to the line position. For an orientation of 1, X refers to the line and Y refers to the character position.

Vital statistics: Entry points: SETCH
 Routines called: NUMARG, PLOTQ
 Common blocks: SKEP

SETCRT--Initialize CRT Beam

Summary: SETCRT will cause the beam to be positioned at (X,Y) without plotting and will set the intensity.

Calling sequence:

CALL SETCRT (X,Y)
 CALL SETCRT (X,Y,I)

where: X is the abscissa at which the beam is to be positioned.
 Y is the ordinate at which the beam is to be positioned.
 I specifies the intensity.
 I=0 for low intensity.
 I=1 for high intensity.

Input arguments: X,Y,I

Output arguments: (none)

Remarks: 1. SETCRT may be called with 2 or 3 arguments.
 2. I=0 initially and will not be reset if omitted.

Vital statistics: Entry points: SETCRT
 Routines called: PLOTQ
 Common blocks: KEEP

SETLCH--Set Type Mode

Summary: SETLCH specifies the starting character position, intensity, case, size, orientation, and type face for the next line plotted by WOT100. Unlike SETCH, the coordinates given here are in the user's own coordinate system and are subject to the scaling given by the current mapping.

Calling sequence: CALL SETLCH
 CALL SETLCH (X,Y)
 CALL SETLCH (X,Y,I)
 CALL SETLCH (X,Y,I,IC)
 CALL SETLCH (X,Y,I,IC,IS)
 CALL SETLCH (X,Y,I,IC,IS,IO)
 CALL SETLCH (X,Y,I,IC,IS,IO,IT)

where:

X,Y	are the coordinates of the center of the first character.
I	specifies the intensity. I=0 for low intensity. I=1 for high intensity.
IC	specifies the case. IC=0 for upper case (normal). IC=1 for lower case.
IS	specifies the size. IS=0 for miniature characters. IS=1 for small characters. IS=2 for medium characters. IS=3 for large characters.
IO	specifies the orientation. IO=0 for horizontal orientation, the line running left to right. IO=1 for vertical orientation, the line running bottom to top.
IT	specifies the type face. IT=0 for Roman type. IT=1 for italic type.

Input arguments: X,Y,I,IC,IS,IO,IT

Output arguments: (none)

Remarks:

1. SETLCH may be called with 0,2,3,4,5,6, or 7 arguments. The call without arguments has its usual effect of using the last set of arguments.
2. I,IC,IS,IO, and IT will be zero initially. Arguments which are omitted will not be reset, unless X and Y are also omitted.

Vital statistics:

Entry points:	SETLCH
Routines called:	NUMARG, PLOTQ
Common blocks:	SMAP, SKEP

SETPCH--Set Plotting Character Parameters

Summary: SETPCH specifies the intensity, case, size, type, face, and spacing of the characters plotted by POINTC and TRACEC.

Calling sequence:

```
CALL SETPCH
CALL SETPCH (I)
CALL SETPCH (I,IC)
CALL SETPCH (I,IC,IS)
CALL SETPCH (I,IC,IS,IT)
CALL SETPCH (I,IC,IS,IT,K)
```

where:

I	specifies the intensity. I=0 for low intensity. I=1 for high intensity.
IC	specifies the case. IC=0 for upper case. IC=1 for lower case.
IS	specifies the size. IS=0 for miniature size. IS=1 for small size.

IS=2 for medium size.
 IS=3 for large size.
 IT specifies the type face.
 IT=0 for Roman type.
 IT=1 for italic type.
 K specifies the spacing.

Input arguments: I, IC, IS, IT, K
 Output arguments: (none)
 Remarks: 1. SETPCH may be called with any number of arguments.
 2. Initially, K=100 and I=IC=IS=IT=0.
 3. K is the minimum number of raster points between characters and must be a nonnegative integer.

Vital statistics: Entry points: SETPCH
 Routines called: NUMARG, PLOTQ
 Common blocks: KEEP

SRDID--Place "SECRET" Message on Film

Summary: SRDID is called by FRAME and CRTID to place the heading "SECRET RD" on microfilm. This routine is not usually called directly.

Calling sequence: CALL SRDID (X,Y)
 where X,Y are the absolute (0.-1023.) coordinates at which the message is to be placed.

Input arguments: X,Y
 Output arguments: (none)
 Vital statistics: Entry points: SRDID
 Routines called: PLOTQ

TRACE--Draw a Solid Line

Summary: TRACE will connect the N points given by the arrays X and Y with straight line segments.

Calling sequence: CALL TRACE (X,Y,N)
 CALL TRACE (X,Y,N,IX)
 CALL TRACE (X,Y,N,IX,IY)

where: X,Y are the names of floating point arrays of X and Y coordinates.
 N is the number of points to be plotted.
 IX, IY are the increments between successive storage locations in X and Y, respectively.

Input arguments: X,Y,N,IX,IY
 Output arguments: (none)
 Remarks: 1. TRACE may be called with 3, 4, or 5 arguments.
 2. IX and IY will be assumed 1 if they are omitted.
 3. See fig. 1 for example.

Vital statistics: Entry points: TRACE
 Routines called: ALOG10, NUMARG, PLOTQ
 Common blocks: SMAP, KEEP

TRACEC--Draw a Solid Line with Label

Summary: TRACEC will connect the N points given by X and Y with line segments and with the specified 280 character superimposed on points more than K raster points apart.

Calling sequence: CALL TRACEC (1HA,X,Y,N)
 CALL TRACEC (1HA,X,Y,N,IX)
 CALL TRACEC (1HA,X,Y,N,IX,IY)

where: A is a single Hollerith character.
 X,Y are the names of floating point arrays of X and Y coordinates.
 N is the number of points to be connected.
 IX, are the increments between successive storage
 IY locations in X and Y, respectively.

Input arguments: A,X,Y,N,IX,IY

Output arguments: (none)

Remarks: 1. TRACEC may be called with 4, 5 or 6 arguments.
 2. The value of K and the size, case, intensity, and type face of the character may be specified in SETPCH.
 3. IX and IY will be assumed 1 if they are omitted.
 4. See fig. 1 for example.

Vital statistics: Entry points: TRACEC.
 Routines called: ALOG10, NUMARG, PLOTQ, SQRT
 Common blocks: SMAP, KEEP

TRACEP--Draw a Dotted Line

Summary: TRACEP will connect the N points given by the arrays X and Y with interpolated points spaced 2 | K | raster points apart.

Calling sequence: CALL TRACEP (X,Y,N)
 CALL TRACEP (X,Y,N,K)
 CALL TRACEP (X,Y,N,K,IX)
 CALL TRACEP (X,Y,N,K,IX,IY)

where: X,Y are the names of floating point arrays of X and Y coordinates.
 N is the number of points to be connected.
 K specifies the spacing and intensity.
 IX, are the increments between successive storage
 IY locations in X and Y, respectively.

Input arguments: X,Y,N,K,IX,IY

Output arguments: (none)

Remarks:

1. TRACEP can be called with 3, 4, 5 or 6 arguments.
2. Interpolated points will always be plotted at low intensity. If K is negative, the data points will be plotted at high intensity; otherwise, the current intensity value will be used.
3. K=+2 initially and will not be reset if omitted.
4. IX and IY will be set to 1 if they are omitted.
5. See fig. 1 for example.

Vital statistics:

Entry points:	TRACEP
Routines called:	ALOGIØ, INTRPL8, NUMARG, PLOTQ
Common blocks:	SMAP, KEEP

WOT1ØØ--Plot Formatted I/O List

Summary: Each record of BCD information, as described by LIST (an ordinary FORTRAN input/output list) and encoded by format NF, will be plotted via a call to PLOTQ. The starting position must be given initially by SETCH or SETLCH.

Calling sequence: ENCODE (L,NF,A) LIST
CALL WOT1ØØ (A,L)

where:

NF	is a FORMAT statement
LIST	is an ordinary FORTRAN I/O list
L	is the number of characters in the output record (L ≤ 136)
A	is variable dimensioned (L+1)/10

Vital statistics:

Entry points:	WOT1ØØ
Routines called:	PLOTQ
Common blocks:	SKEP

SECTION II
UTILITY ROUTINES AND SCRIBE

DYDIM Users Guide

PURPOSE: To allow the user to set up a program with variable array dimensions so that the actual dimensions may be selected conveniently at compile time.

METHOD: DYDIM reads an input file consisting of the desired dimension values followed by the user's code. DYDIM then processes the user's code, inserting the actual dimensions and writing the resulting code on file TZ in a form suitable for compiler input.

CODE PREPARATION: DYDIM will process two types of statements. The first is one in which one or more variable names appear in parentheses (two or more need to be separated by commas) as in Fortran DIMENSION, COMMON, OR EQUIVALENCE statements. The only requirement, however, is that the variable names appear in parentheses. The second type of statement is one in which a variable name appears immediately to the right of an equals sign. The precise format requirements follow.

For variable name(s) in parenthesis, there must be a dollar sign in column one and no blanks imbedded in the variable name. There may, however, be any number of blanks between parentheses, variable names, and commas. Example:

```
$      DIMENSION  X(XDIM), Y(XDIM, YDIM)
```

Note that there are no restrictions on repetition of variable names within a card or entire code.

For a variable name to the right of an equals sign, there must be dollar signs in columns one and two. There may be blanks between the equals sign and the variable name. There must be at least one blank or a dollar sign immediately following the variable name. Neither the blank nor the dollar sign is eliminated by DYDIM.

Example:

\$\$ IX = XDIM * 5 + 3

or

\$\$ IX = 5 * XDIM\$ IY = 10 * XDIM

or

\$\$ XDIM = XDIM

In both types of statements, variable names are limited to seven characters and must not contain imbedded blanks.

INPUT PREPARATION: From the input file (file INPUT by default) DYDIM looks first for a card with a dollar sign anywhere. Finding this, the code scans that card and all cards up to and including the card containing another dollar sign for character strings of the form

NAME = VALUE,

NAME is a variable name of one to seven characters with no imbedded blanks, and VALUE is a one to six digit integer to be inserted in the user's code whenever NAME appears as described above. All parts of the string, including the comma, must appear on the same card, and any number of strings may be placed on a single card. The final string is followed by a dollar sign rather than a comma. Example:

\$ XDIM = 100, YDIM = 83 \$

There must be at least one blank between the first dollar sign and the first variable name.

After these character strings have been read, DYDIM processes the remaining cards on the input file (assumed to be the user's code) for cards as described under code preparation above. Processing continues until an end of file is encountered at which time the file TZ (containing the processed code) is rewound. The output file (OUTPUT by default) contains the quantities NAME and VALUE for each string encountered by DYDIM as described above.

REPETITION OF CARD BLOCKS: DYDIM has the capability of processing groups of cards of the type described above and inserting them at various places throughout the users code. A block of cards is identified with a name and falls between "proc" and "endproc" cards as in the following example.

```
*PROC      NAME1
$    COMMON  X(N1, N2, N3), Y(N1), Z(50)
$    COMMON/BLOCK/A(N1), B(N2)
*ENDPROC
```

The asterisks must be in column one and the name, which may be up to ten characters long, starts in column eleven.

The block of cards is inserted by using an "include" card such as:

```
*INCLUDE      NAME1
```

Again, the asterisk is in column one and the name as above, must start in column eleven.

DECK STRUCTURE: DYDIM required 44000_8 words to compile and executes in 34300_8 words. The time to process a deck varies with the complexity but is about 50 cards per second.

For a program on cards, the deck should be set up as follows:

```
Job Card
Task Card
REDUCE (OFF)
FTN (A)
LGO,
RUN (A, , , TZ)    or    FTN (A, I = TZ)
REDUCE.
LGO.
```

7/8/9

DYDIM deck

7/8/9

\$ NAME1 = VALUE1, NAME2 = VALUE2 \$

*PROC BLOCK1

etc.

*ENDPROC

Program deck

7/8/9

Data Cards

6/7/8/9

For a program on DAFWL, the following setup should be used.

Job Card

Task Card

FTN (A)

DAFWL.

LGO (TAPEZ)

RETURN (TAPEZ)

RUN (A, , , TZ) or FTN(A, I = TZ)

LGO.

7/8/9

DYDIM deck

7/8/9

CARDS

\$ NAME1 = VALUE1, NAME2 = VALUE2 \$

*PROC

etc.

AFWL-TR-74-198

User's program cards

LIBRARY

PLOT0000008

LAST

7/8/9

Data Cards

6/7/8/9

SCRIBE Users Guide

PURPOSE: Given a code number which designates one out of approximately 200 alphabetic, numeric, Greek, or symbolic characters, SCRIBE furnishes a string of x and y coordinates which can be used for plotting the character on the Cal-Comp or CRT.

USAGE: To obtain the plot vector for a given character:

```
CALL SCRIBE (I,J, XMN, XMX, L, VCTR)
```

where I and J are integers defining the character (see below).

SCRIBE returns XMN, XMX, L, and VCTR, where:

XMN and XMX are the left and right edges of the blocks containing the character (used for horizontal spacing)

L is the number of coordinate values in the vector, counting both x and y coordinates: hence $L/2$ is the number of points.

VCTR is the vector (space provided by the user) in which the string of coordinates is placed by SCRIBE. VCTR should be dimensioned to contain about 200 words. (The actual maximum length in the current version is 126, since one of the characters involves 63 points.)

EXAMPLE:

```
PROGRAM TEST ( INPUT,OUTPUT,FILMPL )
DIMENSION VCTR(2,100),XP(100),YP(100)
COMMON /SMAP/ XMN,XMAX,YMIN,YMAX,
1  XMI,XMA,YMI,YMA,XSCALE,YSCALE
CALL PLOTQ(0,0,0,0,7)
XMI=YMI=0.
XMIN=YMIN=0.
XMAX=YMAX=2880.
XSCALE=(XMA-XMI)/(XMAX-XMIN)
YSCALE=(YMA-YMI)/(YMAX-YMIN)
...
```

```

Y=...
CALL SCRIBE (I,J,XMN,XX,L,VCTR)
KMAX = (L/2) + 1
X = X - XMN
M = 0
DO 199 K = 1, KMAX
IF (K .EQ. KMAX) GO TO 198
IF (VCTR (1,K) .GT. 30.9) GO TO 198
M = M + 1
XP (M) = X + VCTR (1,K)
YP (M) = Y + VCTR (2,K)
GO TO 199
198 CONTINUE
IF (M .GT. 0) CALL TRACF (XP,YP,M,1)
M = 0
199 CONTINUE
X = X + XX

```

REMARKS:

1. The plot coordinates are given relative to the center of the character. Therefore Y should be set to the proper value for this position, and X should be set to the left edge of the first character. The statements $X = X - XMN$ and $X = X + XX$ in the above example then advance X to the proper position for plotting the next character.
2. The numbers in VCTR are in floating point form, and all have integral values between -30.0 and +30.0; successive points are joined in a chain of plotted vectors. The value $X = 31.0$ is used to indicate the end of a string of connected vectors.
3. The coordinates furnished by SCRIBE are given in terms of integer values: A change of 1.0 in the value is assumed to represent the smallest increment which can be distinguished by the plotting device.

A character in this example has its top at a larger value of Y than its bottom and its right edge at a larger value of X than its left edge -- i.e., it "faces right." Characters may be plotted "facing up" by writing:

$$\begin{aligned}
 XP (M) &= X - VCTR (2,K) \\
 YP (M) &= Y + VCTR (1,K)
 \end{aligned}$$

and making corresponding changes to place the next character farther along in the Y direction while X is held fixed.

The sizes of the characters are described in the following paragraph, since the size depends upon the "font."

The consequences of applying various scale factors, and of using rotations other than multiples of 90° , have not yet been explored.

CHARACTER DESIGNATION:

Ten "fonts" are provided, corresponding to $I = 1, \dots, 10$ in the example given above. The odd-numbered fonts are large characters. The even numbered fonts are exactly the same characters in a smaller size (and slightly less detail. The characters are illustrated in the accompanying chart.

$I = 1$ or $2:$	$J = 01 - 26:$	Lower case Roman alphabet
	$J = 27 - 36:$	Numbers 0 - 9
	$J = 37 - 48:$	Fortran characters + - * / () \$ = b.,# (note that these are the same as the corresponding 6600 display code, except for #).
$I = 3$ or $4:$	$J = 01 - 26:$	Upper case Roman alphabet
	$J = 27 - 36:$	<u>Sans serif</u> numbers 0 - 9.
$I = 5$ or $6:$	$J = 01 - 26:$	Lower case Greek letters (interpreted according to a special transliteration table: See Chart. Note that \bar{J} and \bar{v} are not used).
$I = 7$ or $8:$	$J = 01 - 26:$	Upper case <u>Greek</u> letters.
$I = 9$ or $10:$	$J = 01 - 61:$	Special characters: See accompanying Chart. Note that the heading on the Chart gives values of J in <u>octal</u> , from 01 to 75.

RESTRICTIONS:

Illegal Characters: Values of I and J which do not appear on the listing above should be avoided. In particular, $I \leq 0$ and $J \leq 0$ may cause dire results. Using a value of J

which is too large for the corresponding value of I will cause a character from some other font to be used, unless J is so large that the "character" is fetched from beyond the entire table.

Space (field length) Requirements: The DAFWL version consists of a source deck which must be compiled by the user. This deck is primarily composed of constants. The subroutine, along with Cal-Comp routines and a fairly short FORTRAN driver, will compile and load in 50000 (octal) field length. It executes in about 30000 (octal) including the microfilm driver.

Time Requirements: Execution time is not excessive -- perhaps 1 second per full page of type.

CHARACTER SET:

The characters used in this subroutine were designed by Mr. Allen V. Hershey of the U.S. Naval Weapons Laboratory, Dahlgren, Virginia.

SCRIBE Chart

Font	01-07	10-17	20-27	30-37
one	abcdefg	hijklmno	pqrstuvw	xyz01234
two	abcdefg	hijklmno	pqrstuvw	xyz01234
three	ABCDEFGFG	HIJKLMNO	PQRSTUVWXYZ	XYZ01234
four	ABCDEFG	HIJKLMNO	PQRSTUVWXYZ	XYZ01234
five	αβχδεφγ	ηι κλμνο	πθρστυ ω	ξψζ
six	αβχδεφγ	ηι κλμνο	πθρστυ ω	ξψζ
seven	ABXΔEΦΓ	HI KΛMNO	ΠΘΡΣΤΤ Ω	ΞΨΖ
eight	ABXΔEΦΓ	HI KΛMNO	ΠΘΡΣΤΤ Ω	ΞΨΖ
nine	∴!?''''°	[]{}<>	±∓×÷≡<>	≤≥α~''''
ten	∴!?''''°	[]{}<>	±∓×÷≡<>	≤≥α~''''

Font	40-47	50-57	60-77
one	56789+-*/()	\$= ,	≠
two	56789+-*/()	\$= ,	≠
three	56789		
four	56789		
five			
six			
seven			
eight			
nine	"" CUCU	E→↑↔↓Δ∇√	∫ϕ∞%&@#§ †‡≡∇×
ten	"" CUCU	E→↑↔↓Δ∇√	∫ϕ∞%&@#§ †‡≡∇×

SECURE Users Guide

PURPOSE AND METHOD: SECURE reads an input file (default is INPUT) then writes a SECRET-RESTRICTED DATA/CNWDI message at the top and bottom of the page along with 59 lines of output (fig. 5). Upon termination SECURE writes a summary page (fig. 6) and a message in the dayfile (fig. 7). In order to terminate properly SECURE must read a double end-of-file. Therefore, a dummy copy must be made (see fig. 7 for example).

THIS RUN TOOK .787 SECONDS AND PROCESSED 428 LINES OF OUTPUT
 DOUBLE END-OF-FILE ENCOUNTERED

```

*****
*
*
* 14 PAGES OF
* UNCLASSIFIED
* RESTRICTED DATA/CNWDI
* ON 10/03/74
* JOB JUL0011 TIME 15.40.04.
*
*
*****
    
```

Figure 6. Example of Summary Page as Given by SECURE

```

10/03/74  SCOPE 3.2.0 - SCM  VER 121  6  010CT74ECPA= 0754
15.39.38.JDL0011
15.39.38.JDL0011,CM52000,1177,P5.
15.39.39.TASK(LETTER10,88091701-9FU,DYS,2007)
15.39.39.FIN(A,R=3,OPT=2,L=SYSOUT,B=SECURE)
15.39.56.      1.317 CP SECONDS COMPILATION TIME
15.39.56.COPY(DUMMY,SYSOUT)
15.39.57.COPY(DUMMY,SYSOUT)
15.40.01.PRESET.
15.40.01.REWIND(SYSOUT)
15.40.01.SECURE(LC=37777,SYSOUT)
15.40.05.*****
15.40.05.*
15.40.05.*
15.40.05.*
15.40.05.*
15.40.05.*
15.40.05.*
15.40.05.*****
15.40.05.*****
15.40.06.END SECURE
15.40.06.PRESET.
15.40.07.CP      3.179 SEC.
15.40.07.PP     10.716 SEC.

```

UNCLASSIFIED
PAGES OF
SECRET - RESTRICTED DATA/CNWDI
PRINTED ON 10/03/74

Figure 7. Example of Dayfile Message as Given by SECURE

DISTRIBUTION

No. cys

1	AUL (LDE), Maxwell AFB, AL 36112
2	DDC (TCA), Cameron Sta, Alexandria, VA 22314
1	AFSWC (HO), Kirtland AFB, NM 87115
	AFWL, Kirtland AFB, NM 87117
2	(SUL)
15	(DYS)
1	(ADPT)
1	Official Record Copy (Dr. Rhoades, DYS)

THIS REPORT HAS BEEN DELIMITED
AND CLEARED FOR PUBLIC RELEASE
UNDER DOD DIRECTIVE 5200.20 AND
NO RESTRICTIONS ARE IMPOSED UPON
ITS USE AND DISCLOSURE.

DISTRIBUTION STATEMENT A

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.